Bilexical Grammars and a O(n^3) Probabilistic Parser

Jason Eisner
University of Pennsylvania
IWPT - 1997

Soft Selection

doff a cap
sombrero
shirt
sink
clothe
about
...
Adjuncts too: doffed his cap to her at her for her

Lexicalized Grammars

doff: ___ NP
doff: (S/NP)/NP
S → NP doff NP

Rules are specialized for individual words
(or are implicit in lexical entries)

From lexical to bilexical

Lafferty et al. 92, Charniak 95, Alshawi 96, Collins 96, Eisner 96, Goodman 97
Also see Magerman 94, Ratnaparkhi 97, etc.

Rules mention two words
E.g., each verb can have its own distribution of arguments

Goal: No parsing performance penalty
Alas, with standard chart parser:
nonlexical O(n^3)
lexical O(n^5)
other methods give O(n^4) or O(n^3)
bilexical O(n^5)

Simplified Formalism (1)

The cat in the hat wore a striped stovepipe to our house today.

(save these gewgaws for later)
Need a flexible mechanism to score the possible sequences of dependents.

### Weighting the Grammar

- **doff**: right DFA
  - Transitive verb. accepts: Noun Adv
- **likes**: hat nicely now (e.g., "Bentley doffed [his hat] nicely [just now]")
- **hates**: sink countably (e.g., "Bentley doffed [the sink] countably")

#### Simplified Formalism (2)

```
  wore
  cat
  stovepipe
  to
  today
```

- **wore**: left DFA
- **right DFA**
  - cat
  - stovepipe to today

### Why CKY is slow

1. visiting relatives is boring
2. visiting relatives wear funny hats
3. visiting relatives, we got bored and stole their funny hats

- **visiting relatives**: NP(visiting), NP(relatives), AdvP, ...

  CFG says that all NPs are interchangeable
  So we only have to use generic or best NP.

  But bilexical grammar disagrees:
  e.g., NP(visiting) is a poor subject for wear
  We must try combining each analysis w/ context

### Generic Chart Parsing (1)

- interchangeable analyses have same signature
  - "analysis" = tree or dotted tree or ...
    - ... (cap spending at $300 million) ...
    - VP [score: 17]
    - VP [score: 12]
    - NP [score: 8]
    - [score: 4]

  1. if \( \leq S \) signatures, keep \( \leq S \) analyses per substring

### Generic Chart Parsing (2)

- for each of the \( O(n^2) \) substrings,
  - for each of \( O(n) \) ways of splitting it,
    - for each of \( \leq S \) analyses of first half
      - for each of \( \leq S \) analyses of second half,
        - for each of \( \leq cS \) ways of combining them:
          - combine, & add result to chart if best

<table>
<thead>
<tr>
<th>cap spending</th>
<th>[at $300 million]</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \leq S ) analyses</td>
<td>( \leq S ) analyses</td>
</tr>
<tr>
<td>( \leq S ) analyses</td>
<td>( \leq cS ) analyses</td>
</tr>
<tr>
<td>of which we keep ( \leq S )</td>
<td></td>
</tr>
</tbody>
</table>

### Headed constituents ...

... have too many signatures.

**How bad is \( \Theta(n^3S^2c) \)?**

For unheaded constituents, S is constant: NP, VP ...
(similarly for dotted trees). So \( \Theta(n^3) \).

But when different heads \( \Rightarrow \) different signatures,
the average substring has \( \Theta(n) \) possible heads
and \( S=\Theta(n) \) possible signatures. So \( \Theta(n^2) \).
Forget heads - think hats!

Solution:
Don’t assemble the parse from constituents.
Assemble it from spans instead.

Spans vs. constituents
Two kinds of substring.
» Constituent of the tree: links to the rest only through its head.
  The cat in the hat wore a stovepipe. ROOT
  The cat
  in the hat
  wore a stovepipe. ROOT

» Span of the tree: links to the rest only through its endwords.
  The cat in the hat wore a stovepipe. ROOT

Decomposing a tree into spans
The cat in the hat wore a stovepipe. ROOT
The cat in the hat wore a stovepipe. ROOT
The cat wore a stovepipe. ROOT
The cat in the hat wore a stovepipe. ROOT

Maintaining weights
Seed chart w/ word pairs $x\ y, x\ z, y\ z$
Step of the algorithm:
$$a\ ...\ b\ b\ ...\ c = a\ ...\ b\ ...\ c$$
We can add an arc only if $a, c$ are both parentless.
$$\text{weight}(a\ ...\ b\ ...\ c) = \text{weight}(a\ ...\ b) + \text{weight}(b\ ...\ c)$$
+ weight of c arc from a’s right DFA state
+ weights of stopping in b’s left and right states

Analysis
Algorithm is $O(n^3 S^2)$ time, $O(n^2 S)$ space. What is $S$?
$$a\ ...\ b + b\ ...\ c = a\ ...\ b\ ...\ c$$
Where:
  $b$ gets a parent from exactly one side
  Neither $a, c$ previously had a parent
  $a$’s right DFA accepts $c$; $b$’s DFAs can halt

Signature of $a\ ...\ b$ has to specify
parental status & DFA state of $a$ and $b$

$\therefore S = O(t^2)$ where $t$ is the maximum # states of any DFA
$S$ independent of $n$ because all of a substring’s analyses are headed in the same place - at the ends!

Improvement
Can reduce $S$ from $O(t^2)$ to $O(t)$
$$a\ ...\ b + b\ ...\ c = a\ ...\ b\ ...\ c$$
State of $b$’s left automaton tells us weight of halting
Likewise for $b$’s right automation

Add every span to both left chart & right chart
Above, we draw $a\ ...\ b$ from left chart, $b\ ...\ c$ from right chart
Copy of $a\ ...\ b$ in left chart has halt weight for $b$ already added
so its signature needn’t mention the state of $b$’s automation
Embellishments

1. More detailed parses
   - Labeled edges
   - Tags (part of speech, word sense, ...)
   - Nonterminals

1. How to encode probability models

More detailed parses (1)

Grammar:
- DFAs must accept strings of word-role pairs e.g., (cat, agent) or (hat, obj)
- When we stick two spans together, consider covering with: nothing, agent, agent, obj, obj, etc.

Parser:
- Time penalty: \(O(m)\) where \(m\) is the number of label types

More detailed parses (2)

Optimize tagging & parsing at once

Grammar:
- Every input token denotes a confusion set. e.g., cat = \{cat1, cat2, cat3, cat4\}
- Choice of cat adds a certain weight to parse

Parser:
- More possibilities for seeding chart
- Tags of b must match in \((a, b)\) \& \((b, c)\)
- Signature of \((a, b)\) must specify tags of a and b

Time penalty:
- \(O(g^4)\) where \(g\) is max # tags per input word
- \(O(g^3)\) by considering only appropriate \(a \cdots b\) \& \(b \cdots c\)

Nonterminals

Articulated phrase
- One-to-one (cf. Collins 96)
- Use fast bilexical algorithm, then convert result to nonterminal tree.

Want small (and finite) set of tags like \(A, B, C\).
(Guaranteed by X-bar theory: doff = \{doffV, V, VP, doffV, V, VP, S\}.)

String-local constraints

Seed chart with word pairs like \((x, y)\)

We can choose to exclude some such pairs.

Example: k-gram tagging. (here \(k=3\))

Tag with part-of-speech trigrams

Weight = \(\log Pr(\text{the} | \text{Det} | \text{N,P})\)

Exclude bigram:

The 2 words disagree on tag for “cat”
Conclusions

- Bilexical grammar formalism
  - How much do 2 words want to relate?
  - Flexible: encode your favorite representation
  - Flexible: encode your favorite prob. model

- Fast parsing algorithm
  - Assemble spans, not constituents
  - \(O(n^3), \text{not } O(n^5)\).
  - Precisely, \(O(n^{3t^2g^3m})\).
  - \(t=\max \text{ DFA size, } g=\max \text{ senses/word, } m=\# \text{ label types}\)
  - These grammar factors are typically small